Preprint
UCRL-JC-143820

Tracking and Motion Analysis of Crack Propagations in Crystals for Molecular Dynamics

L. V. Tsap, M. Duchaineau and D. B. Goldgof

This article was submitted to
IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, December 11-13, 2001

May 14, 2001

U.S. Department of Energy
Lawrence Livermore National Laboratory

Tracking and Motion Analysis of Crack Propagations in Crystals for Molecular Dynamics

Leonid V. Tsap, Mark Duchaineau
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA 94551
{tsap, duchaine}@llnl.gov

and Dmitry B. Goldgof
Dept. of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
goldgof@csee.usf.edu

Abstract [1]

This paper presents a quantitative analysis for a discovery in molecular dynamics. Recent simulations have shown that velocities of crack propagations in crystals under certain conditions can become supersonic, which is contrary to classical physics. In this research, we present a framework for tracking and motion analysis of crack propagations in crystals. It includes line segment extraction based on Canny edge maps, feature selection based on physical properties, and subsequent tracking of primary and secondary wavefronts. This tracking is completely automated; it runs in real time on three 834-image sequences using forty 250 MHZ processors. Results supporting physical observations are presented in terms of both feature tracking and velocity analysis.

1 Introduction

Molecular dynamics involves solving Newton's laws of motion for large numbers of atoms (millions to billions) while obeying classical force laws. Crystal deformations under stress are strongly dependent on the movement of cracks (dislocations). So far, the dynamics of dislocations is not well understood because of the complexity and the difficulties in modeling the phenomena [1]. Cracks direct macroscopic object energy down to breaking bonds at the atomic scale [10]. Therefore, atomic studies are important for the development of the crack propagation theory and applications which range from studies of composite man-made materials to earthquake analysis. Such large-scale simulations have become possible recently due to large increases in computing power, thereby allowing studies of the dynamics of dislocations in a consistent way. Recent computer large-scale simulations in molecular dynamics challenge previous beliefs that cracks cannot propagate faster than the Rayleigh wave speed. Conventional views are based on continuum mechanics, including dynamic elastic solutions and energy release rates.

Our goal is to develop a completely automated tracking and analysis system that measures speed and examines behavior of dislocations. This paper presents our image analysis approaches for detecting and selecting features of interest from moderately noisy data, which is described next. The system includes line segment extraction based on Canny edge maps, feature selection based on physical properties, and subsequent tracking of primary and secondary dislocations.

A number of recent approaches included a primitive generation step using line extraction. Borges and Aldon [2] achieved line extractions with a prototype-based fuzzy clustering algorithm in a split-and-merge framework. Van der Heijden [19] describes line features in terms of covariance functions. Matching junctions based on uncertainty propagation from the detection to the grouping stage was presented by Shen and Palmer [15]. Ji and Haralick [6] also used uncertainties for an optimal Hough transform for line detection. Nandy and Ben-Arie [11] applied an expansion-matching method while treating junctions as combinations of elementary features. Line extraction in noisy images was described by Lee and Kweon [8].

Line extraction is a pre-processing step in a feature recognition and structure analysis for many successful applications. Kim *et al.* [7] produced building descriptions from multiple images by matching lines and junctions for 3-D feature generation. Price [14] addressed the problem of extracting street grids in an urban environment by matching intersection features. Other notable applications include Magnetic Resonance Imaging (MRI) [9] and aerial imagery [12].

2 Description of the Method

2.1 Data: from Atoms to Pixels

The data represents molecular dynamics simulation of crack propagations along a weak interface joining two crystals [1] with different material properties. Dislocation is caused by shear stress, which is produced

[1] This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48. UCRL-JC-143820

by moving top and bottom elements in different directions. The simulation continues until dislocation spreads to the right boundary. Each sequences contain 834 frames of 3100x790 images.

After propagating (left to right) initially at the expected subsonic speed, the crack tip increases its velocity (Figure 1). Creation of the daughter crack denoted as (2) is shown Figure 1(b) where the mother crack is denoted as (1) in both figures. The daughter crack is now associated with the increased (supersonic) velocity of the rightmost crack tip, while the mother crack keeps traveling at the old speed.

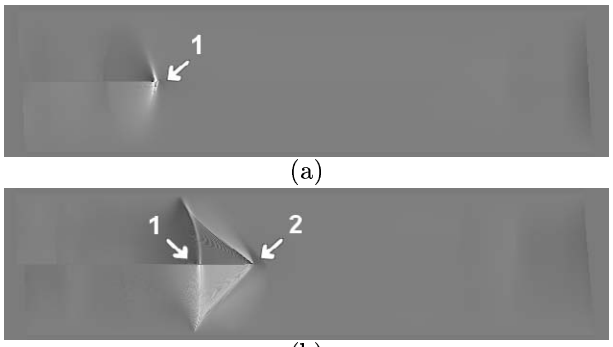


Figure 1: Crack transition to a supersonic speed.

Association between atoms and pixels is as follows. Each atom has coordinates x, y, z, field values (velocity in our case) vx and vy, and a potential energy *epot* value. Simulations range from 20 million to 1 billion atoms. Typical analysis is done on 500-3000 snapshots in time. In the simulation providing data for this research, we are looking at a 2-D slice of a very thin 3-D problem containing around 20 million atoms. The atoms during a simulation run are used to "paint" a 3-D regular grid by using weighted averaging with a pseudo-Gaussian kernel (Figure 2(b))

$$w = \begin{cases} \frac{1}{1+\frac{9r^2}{R^2}} - 0.1 & if r < R \\ 0 & otherwise \end{cases} \quad (1)$$

where R is the specified atom radius, and r is the distance between the atom and pixel centers.

For every point on a pixel grid, its field value at a pixel f_p (visualized in greyscale in displayed images) represents weighted velocity of selected i atoms and is computed as (Figure 2(a))

$$f_p = \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i} \quad (2)$$

In Figure 2(a) atoms are represented as circles on a pixel grid. Their centers are connected with pixels which are located within respective R.

The weights in boundary regions are kept to facilitate distributed parallel "painting" and wavelet compression with no node-to-node communication. Weights are subsequently used during post-process analysis to form complete seamless regular grids of vx, vy, *epot* fields. The regular grid spacing is chosen to be the same as the average inter-atom spacing in the crystal being simulated.

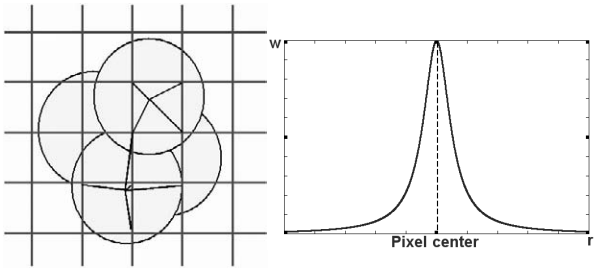


Figure 2: Weighted-averaging of atom field values using pseudo-Gaussian profile.

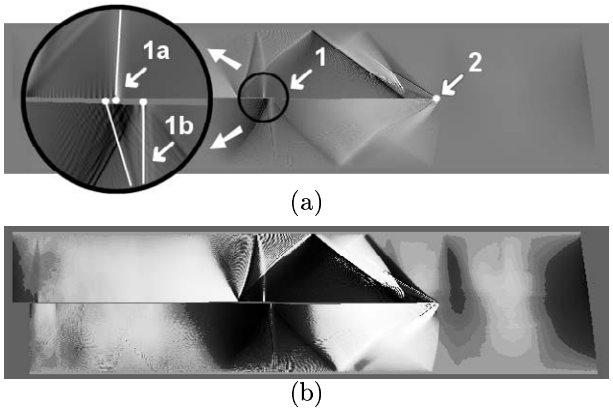
The following sections describe steps needed for the edge, line, and feature extraction, followed by tracking of the parent and child dislocations, and their motion analysis.

2.2 Edge Detection

One of the later frames in the sequence is shown in Figure 3(a). Similarly to Figure 1, feature (1) in Figure 3 denotes the mother crack while (2) is the rightmost cracktip (the daughter crack). An enlarged circle zooms in on the differences in crack propagation in top (1a) and bottom (1b) materials. White lines indicate wavefront features, while white points represent their intersections with the material boundary ("starting points" for cracks). All of these features need to be tracked for "composite" features, such as (1b), an average position is an adequate measure of crack propagation.

The histogram equalization in Figure 3(b) shows potential problems for a meaningful edge detection, where "meaningful" means selected features required for automated tracking. There is considerable background noise and a number of irrelevant features created by shock waves bouncing off the boundaries. In this application it is better to include noise than to miss a real feature.

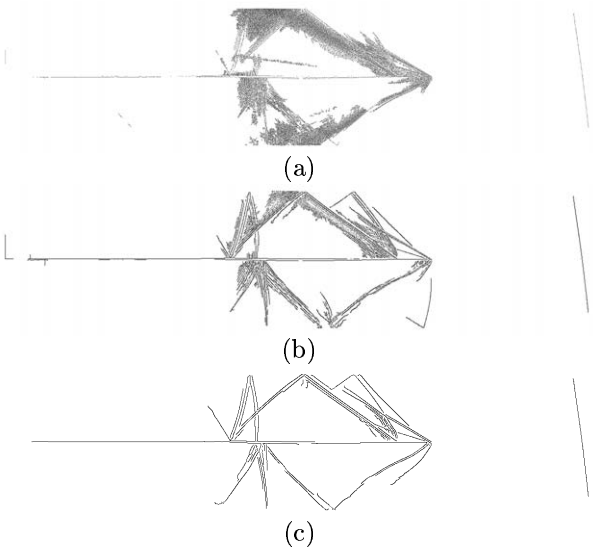
A number of edge detectors have been tested; and results have been compared with ground truth. Canny edge detector [3] produced the best results in terms of selecting elements needed for the subsequent line and feature detectors. Canny edge detection includes smoothing by Gaussian convolution, followed by an application of a simple 2-D first derivative operator for computing the pixel-wise edge strength and direction. Different scales for the operator are represented by different standard deviations of the Gaussian smoothing filter, σ. Subsequent non-maximal suppression yields



(a)

(b)

Figure 3: A frame with a complex set of features. (a) Magnification shows differences in the mother crack propagation in two materials. (b) Histogram equalization shows potential problems for a meaningful edge detection.



(a)

(b)

(c)

Figure 4: Parameter and scale selection for edge detection.

thin edge lines in the output (which is important for an effective line extraction and feature detection described in following sections). Hysteresis thresholding is used to decrease the probability of breaking noisy edge output into multiple fragments. Although edge tracking starts with pixels above a high threshold T_{high} (certain edge pixels), points above a low threshold T_{low} could still belong to edges if they are connected to certain edge pixels. Canny edge detector code (developed by Heath *et al.* [4]) is publicly available [17].

Parameter selection $(\sigma, T_{low}, T_{high})$ is crucial since we cannot miss important features (Figure 1) or dominate them with noise (Figure 3). Application of the best fixed parameter set (0.60, 0.30, 0.90) as discussed in [4] is shown in Figure 4(a). Important features are lost in the noise. A change in scale to 1280x326 helps somewhat, but contours of cracks and shock wave reflections are still distorted (Figure 4(b)). The optimal parameter set for our images (1.50, 0.10, 0.93) produces satisfactory results (Figure 4(c)). Increasing σ reduces the detector's sensitivity to noise, which results in reduced numbers of lines representing wavefronts. T_{low} is decreased to prevent noisy edges from breaking up. Setting T_{high} slightly higher lessens the number of spurious edge fragments. The selected parameter set is capable of selecting required attributes even at a 640x163; however, such scale leads to data underutilization and to a somewhat lower precision of motion analysis. Of course, fine scales yield noise and an increase computation time. Experimentally, 1280x326 is shown to be the best compromise when considering such criteria.

2.3 Line Extraction

During this step, entities constituting edges are identified, and adjacency relationships of resulting line segments are determined. First, connected components labeling is performed recursively by scanning through the image across each row and linking 8-connected [18] pixels with edge values:

- if the pixel has no connected edge pixels, then a new unique label is assigned, and a new chain is started;
- if such pixels exist and share the same label, it is assigned to a pixel in question; and
- multiple labels in connected pixels are resolved by the subsequent pass.

Resulting chains are split into distinctive segments at high curvature points (if the curvature has been increasing for previous pixels) and by the polyline splitting [5]. Implementation is based on the program by Shin *et al.* [16]. A straight-line model $y(x) = ax + b$ is fitted to each contour segment using least squares with possible errors in both coordinates [13]. Then, endpoints are projected onto lines to define resulting line segments. For each endpoint i with coordinates (x_i, y_i), its respective projection (x'_i, y'_i) satisfies the line equation as $y'_i = ax'_i + b$. Also, the product of the gradients of perpendicular lines can be computed as

$$a\frac{y'_i - y_i}{x'_i - x_i} = -1 \qquad (3)$$

Hence, by substituting y'_i in (3), we obtain

$$x'_i = \frac{a}{a^2+1}(y_i + \frac{1}{a}x_i - b) \qquad y'_i = ax'_i + b \qquad (4)$$

Lines shorter than an adaptive threshold, empirically determined to be 1/64 of the image width at a current scale, are excluded from further processing. We do not want to make it more restrictive and miss crack features, which in some frames (when "reflections" from a daughter crack overlap a parent crack

feature) could be quite small. Therefore, choosing the right edge detector parameter set and scale (Figure 4), as described in the previous section, is very important.

2.4 Feature Detection, Modeling and Correspondences

There are several types of features of interest. First, there is a cracktip denoted as (1) in Figure 1(a) and as (2) in Figure 1(b) (numbering in the parent-child order, not in the feature type order). Second, the parent crack gradually splits into two which propagate at different rates ((1) in Figure 1(b) or even more obvious as (1a) and (1b) in Figure 3(a)) since top and bottom crystals have different material properties. Localization of the latter two features on the structural level is based on observations that they are relatively short lines almost perpendicular to the longest horizontal line in the image. The system has to detect parts of these features and model them with line segments. Coordinates of their intersections with the material boundary (the central horizontal line) provide the necessary information for the estimation of velocities of dislocations, and pinpoint jumps to supersonic speeds. The search for the two sections of the mother crack begins by locating candidate points within 5% of image height from the material boundary. For each of these points, respective lines are evaluated whether they are within 20 degrees ($\pi/9$) from perpendicular to the material boundary:

$$||arctan\frac{y_2 - y_1}{x_2 - x_1}| - \frac{\pi}{2}| < \frac{\pi}{9} \qquad (5)$$

As a result, two groups of line segments (above and below the material boundary) are identified. An average, representing wavefront propagation in a given material, is computed for each of these groups. We get some spurious responses due to noise and other features present in images. However, they are always outside regions where true features are expected. Estimations are based on the history of the respective feature propagation, and are updated after every detection. This allows to separate features from erroneous responses. This produces an automatic feature detection for the parent crack.

What about the main cracktip? The algorithm responsible for cracktip detection scans from right to left for the first column of pixels that has values in the top and bottom 10 percentile of the entire histogram, which is computed after the enhancement of horizontal edges. It filters out all the irrelevant intersections by starting at the known crack tip initial position and tracking the subsequent intersections with constraints on the expected velocities and an estimated maximum line-feature width. This algorithm operates on 3100x790 full-size images.

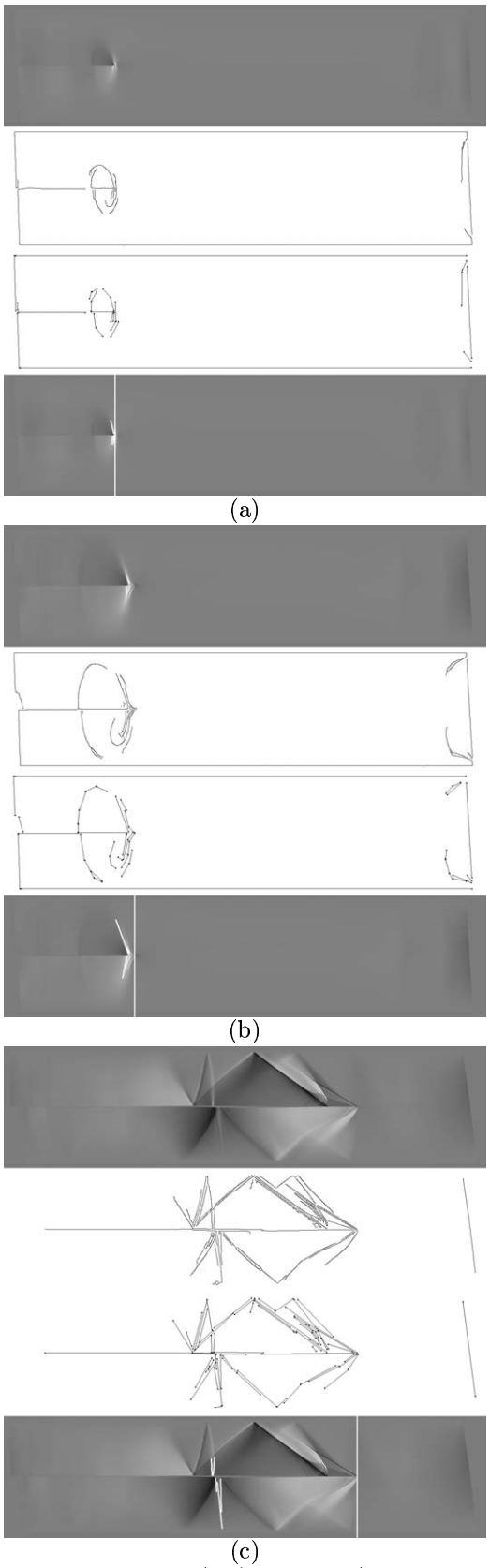


Figure 5: Each figure (a-c) includes (top to bottom): original frames, results of the edge detection, line extraction, and feature tracking.

Outputs of the two algorithms are combined at a finest scale. Thus, feature detection across the sequence proceeds automatically. Establishing correspondences becomes a trivial process since we have three distinct features in every frame: the main cracktip and two features representing a parent wavefront.

3 Experimental Results and Motion Analysis

The system runs on a SGI machine with fourty-eight 250 MHZ IP27 processors and 15872 Mbytes of main memory. We explicitly allocate 40 processors via a parallelization tool where a script is run independently on each work unit. In the case of this study, a work unit is typically a timestep (i.e., a frame in the image sequence).

Figures 5(a-c) include selected original frames (top to bottom), results of the edge detection, line extraction, and feature tracking. Feature tracking is indicated by:

- vertical lines tracking cracktips for mother and daughter dislocations in both materials, and
- relevant features representing two parts of the mother crack propagating at different speeds.

Figure 5(a) represents an initial state with a single wavefront, the creation of a daughter crack is shown in Figure 5(b), and one of the later timesteps is reflected in Figure 5(c).

We allowed for false matches; however, using prediction based on the recent propagation history allows the localization of a region where a correct match is expected. The third image in each of Figures 5(a-c) contains some erroneous data obviously located outside regions of interest. When false features outside such regions are discarded, there are no false alarms left at all. In each 834-frame sequence, mother crack features were present in 684 frames. For the entire 834-frame sequence, there were 78 instances needing interpolation for the upper part of the mother crack, and 37 misses for the lower.

Displacement and velocity functions for features of interest are shown in Figures 6(a) and (b), respectively. Both graphs have time in the X direction, with timesteps changing from 1 to 834 on the far right. Similarly to Figure 3(a), (2) denotes the daughter crack, while (1a) and (1b) represent two parts of the mother crack propagating in above and below the material boundary, respectively. The velocity plot allows us to pinpoint intervals when the speed of the rightmost cracktip drastically jumps, thus bringing it to the supersonic speed. These two peaks indicate the emergence of the daughter and granddaughter cracks, respectively.

Velocity is computed as

$$V_i = \frac{P_{i+2} - P_{i-2}}{t_{i+2} - t_{i-2}} \tag{6}$$

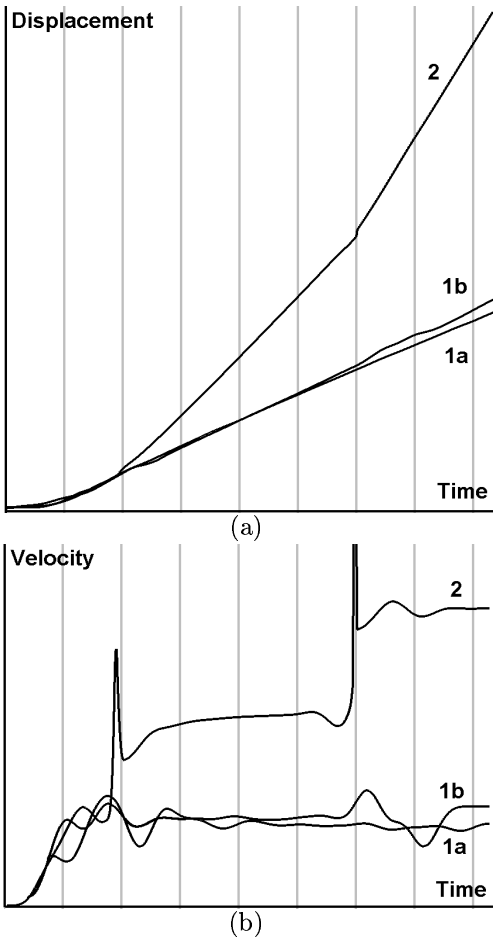


Figure 6: Displacement and velocity functions for features of interest.

where P_i is a position of the feature (an x coordinate in our case), and the denominator is essentially a constant.

This constant is very small since we need to distinguish the emergence of events and keep significant features while avoiding discontinuities. This allows for increased effects of noise. To prevent such occurrences in the velocity plots, a smoothing procedure is used to diffuse second differences. The smoothing is by positions, but it is designed to affect velocity plots. It finds the smoothest curve that varies no more than the specified maximum pixel error from the original unsmoothed data. This constrained smoothing is performed by iteratively averaging out (diffusing) the second differences d_i of the position arrays P_i and clamping values back to within the constraints as needed:

$$d_i = P_i - \frac{P_{i+1} + P_{i-1}}{2} \tag{7}$$

$$P_i' = P_i + q(\frac{d_{i-1} + d_{i+1}}{2} - d_i) \tag{8}$$

$$P_i'' = \begin{cases} P_i' & if P_i' \in [P_i - e, P_i + e] \\ P_i - e & if P_i' < P_i - e \\ P_i + e & otherwise \end{cases} \tag{9}$$

where q is a step size, e is a maximum pixel error, and P_i' and P_i'' represent updates before and after applying the constraint, respectively.

Note that the constrained smoothing for the crack-tip positions assumes a pixel error e of +/- 4 pixels with respect to the full-resolution (3100x790) images. For the transverse-wave-speed intersection positions, the error assumption increases to +/- 16 pixels because of the change in scale in that detection process.

4 Summary and Conclusions

In this paper we presented a completely automated tracking and analysis system that measures speed and examines behavior of dislocations. It provides the basis for a quantitative analysis for a discovery in molecular dynamics. Recent simulations have shown that velocities of crack propagations in crystals under certain conditions can become supersonic, which is contrary to classical physics. In this research, we presented a framework for tracking and motion analysis of crack propagations in crystals. It includes line segment extraction based on Canny edge maps, feature selection based on physical properties, and subsequent tracking of primary and secondary dislocations. This tracking is completely automated; it runs in real time on three 834-image sequences using forty 250 MHZ processors. Results supporting physical observations were presented in terms of both feature tracking and velocity analysis. Molecular dynamics simulations and their quantitative analysis have shed light on the question of whether and how a crack undergoes the transition from subsonic to intersonic velocities.

Acknowledgment

We would like to thank Farid F. Abraham from IBM Almaden Research Center who developed supersonic cracks studies. This work was done as a part of LLNL VIEWS and SAVAnTS visualization projects.

References

[1] F. F. Abraham and H. Gao. How fast can cracks propagate? *Physical Review Letter*, 84(14):3113–3116, 2000.

[2] G. A. Borges and M.-J. Aldon. A split-and-merge segmentation algorithm for line extraction in 2-D range images. In *Proc. ICIP*, pp. 441–444, Barcelona, Spain, September 2000.

[3] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on PAMI*, 8(6):679–698, November 1986.

[4] M. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer. A robust visual method for assessing the relative performance of edge detection algorithms. *IEEE Trans. on PAMI*, 19(12):1338–1359, December 1997.

[5] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, New York, 1995.

[6] Q. Ji and R. M. Haralick. An optimal bayesian hough transform for line detection. In *Proc. ICIP*, vol. 2, pp. 691 – 695, Kobe, Japan, October 1999.

[7] Z. W. Kim, A. Huertas, and R. Nevatia. Automatic description of complex buildings with multiple images. In *IEEE Workshop on Applications of Computer Vison*, pp. 155 – 162, Palm Springs, CA, December 2000.

[8] J. W. Lee and I. S. Kweon. Extraction of line features in a noisy image. *Pattern Recognition*, 30(10):1651–1660, October 1997.

[9] G. Lohmann and F. Kruggel. Extracting lines of maximal depth from mr images of the human brain. In *Proc. ICPR*, vol. 3, pp. 518 – 522, Vienna, Austria, August 1996.

[10] M. Marder. Molecular dynamics of cracks. *Computing in Science and Engineering*, 1(5):2–9, September/October 1999.

[11] D. Nandy and J. Ben-Arie. Generalized feature extraction using expansion matching. *IEEE Trans. on Image Processing*, 8(1):22 – 32, January 1999.

[12] R. L. Pires, P. De Smet, and I. Bruyland. Line extraction with the use of an automatic gradient threshold technique and the hough transform. In *Proc. ICIP*, vol. 3, pp. 909 – 912, Vancouver, BC, Canada, September 2000.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, MA, 1993.

[14] K. Price. Urban street grid description and verification. In *IEEE Workshop on Applications of Computer Vison*, pp. 148 – 154, Palm Springs, CA, December 2000.

[15] X. Shen and P. Palmer. Uncertainty propagation and the matching of junctions as feature groupings. *IEEE Trans. on PAMI*, 22(12):1381 – 1395, December 2000.

[16] M. C. Shin, D. B. Goldgof, and K. W. Bowyer. An objective comparison methodology of edge detection algorithms using a structure from motion task. In *Proc. CVPR*, pp. 190–195, Santa Barbara, CA, June 1998.

[17] Edge Detector Performance Evaluation Study. *ftp://figment.csee.usf.edu/pub/Edge_Comparison/source_code/canny.src*.

[18] S. E. Umbaugh. *Computer Vision and Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1998.

[19] F. van der Heijden. Edge and line feature extraction based on covariance models. *IEEE Trans. on PAMI*, 17(1):16–33, January 1995.